

## **Appendix B**

### **Concepts used in an Exemplary Object Model or Object Class Definitions**

For the purpose of describing different concepts, the examples of (Java) class definitions given in the Figure 19 use Java. These classes have been defined using the sample database 'pubs' which comes with Microsoft SQL Server relational database management system.

#### **Primitive and complex attributes**

A primitive attribute of a class is an instance variable of simple (primitive) type. For example, attributes of type integer, String, boolean are primitive attributes. In the Figure 19, title\_id and ytd\_sales of class Title are primitive attributes.

A complex attribute is a reference to an object (or a collection of objects) of another class. For example pubInfo and titles of class Publisher are complex attributes.

In general, all instances of a class are stored in the same relational table. Primitive attributes are stored in the columns of that class. The appropriate SQL data type of the column may be different. However, a complex attribute is not (cannot be) stored in the same table. Instead, the object(s) referenced by a complex attribute is stored in the table of the corresponding class. To subsequently retrieve those referenced objects as part of the containing object, a set of attributes of the containing object is used (as foreign key) to locate them. If a complex attribute is referencing a collection of objects, their ordering may be specified.

## Containment by value

All primitive attributes are contained by value in an object. That is, they cannot exist independent of the object. Objects referenced by a complex attribute that cannot logically exist without the existence of the referencing (containing) object are also said to be contained by value. If the containing object is deleted, the contained by value objects should also be deleted. pubInfo and titles are examples of attributes contained by value in the class Publisher.

## Containment by reference

A complex attribute which points to an object which may exist independent of the existence of the referencing (containing) object is an attribute contained by reference. If the referencing (containing) object is deleted, the referenced object need not be deleted. The publisher attribute of Emp class is contained by reference.

## Collection class

A collection class is a pseudo-class describing a certain type (array or vector) of collection of objects of a given class. A collection class has an optional notion of ordering of elements of the collection. Collection class specification is needed to describe those complex attributes which reference a collection (array or vector) of objects. This is needed only for object relational mapping specification; no explicit class is defined at the language level.

## Shallow and Deep queries

When an object is normally retrieved from the database, it is retrieved with all the complex attributes instantiated with referenced objects (recursively). This is called deep query. If none of the referenced objects are retrieved (i. e. only the primitive attributes are retrieved) , it is called a shallow query.

## Directed operation options

Sometimes, just deep or shallow options may not be the most optimal way of doing an operation (query, insert, update, delete). For example, if an object has 5 complex (reference) attributes and we are interested in getting just 2 of them, how do we specify that? The answer is directed options. Directed options further qualify the depth of the operation in one of the following ways: 1) if the overall operation is deep, one may specify a list of class and attribute name pairs such that all except those complex attributes should be followed and 2) if the overall operation is shallow, one may specify a list of class and attribute name pairs such that only those complex attributes should be followed. Depending upon the application needs at certain stage of processing, these options may be employed to achieve efficiency or some semantic requirements.

## Database URL

A database URL contains the URL (Universal Resource Locator) understood by the appropriate JDBC driver. Additionally, it has user name and password components.

- 1 Optionally, it has information about Object-Relational Mapping Id (ORMId) and/or
- 2 Object-Relational Mapping File (ORMFile). They have been explained further under
- 3 the grammar rules of <DATABASE-URL> in a subsequent section.